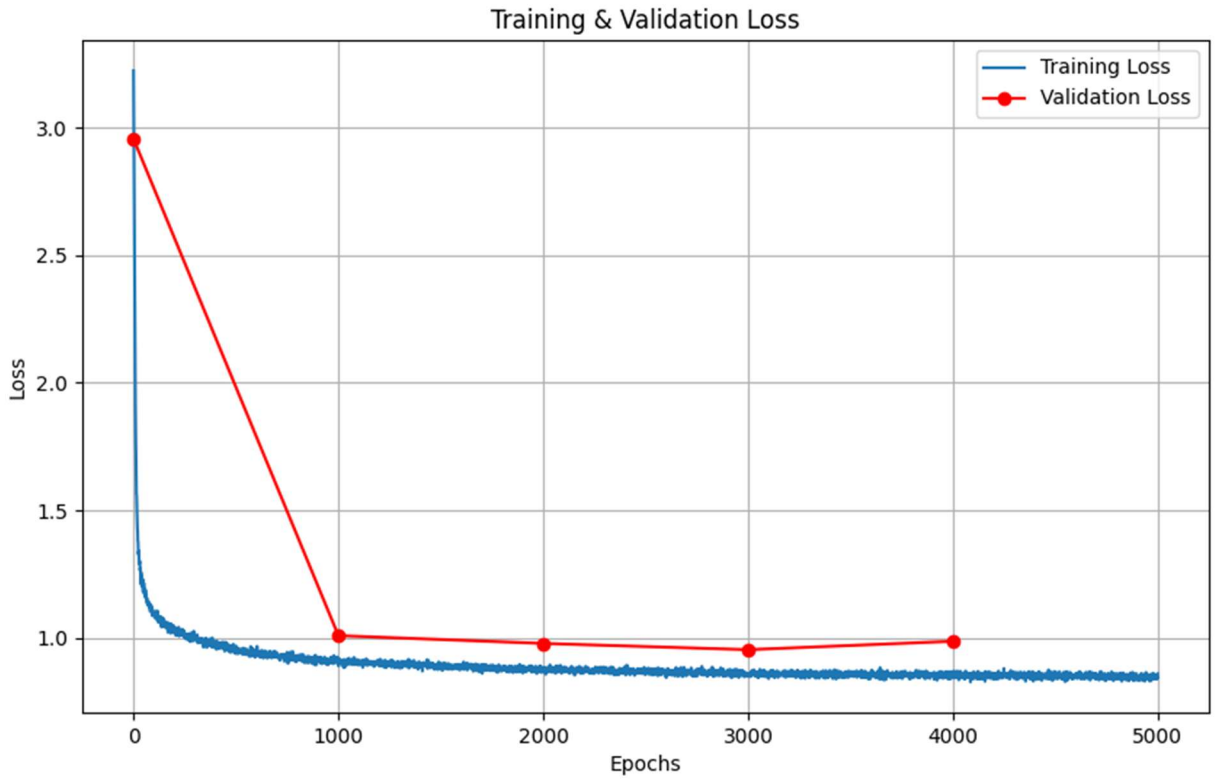
My neural network is relatively straightforward consisting of an input layer, one hidden layer and an output layer. I was able to get good performance using the default 16 hidden neurons but found that increasing the epochs allowed the model to converge to a better local minimum.

To get an idea of how much the model was overfitting, I split the data into not only train and test, but also validation. Every 1000 epochs, the model was evaluated on the validation set and if the loss began to increase, I knew the model was overfitting and should modify my parameters. I experimented with using dropout as a form of regularization while adding more hidden neurons, but I was not able to get any significant performance improvements.

Additionally, I experimented with scheduling to reduce the learning rate after an initial warm up period. Empirically this made some improvements, but they weren't very significant.

In the end, my model was able to achieve a test set accuracy of ~60% which is good considering the null accuracy is just 16.7%




```
Epoch 19800, Loss: 0.8123, Learning Rate: 0.000007
Epoch 19900, Loss: 0.8257, Learning Rate: 0.000007
Test Loss: 1.0597
Test Accuracy: 0.6042
```